AD 703 561

INFORMATION SYSTEM DESIGN IN LARGE SCALE LOGISTIC SYSTEMS

E. P. Durbin

Rand Corporation
Santa Monica, California

March 1970

AD703561

INFORMATION SYSTEM DESIGN IN LARGE SCALE LOGISTIC SYSTEMS

E. P. Durbin

March 1970

P-4308

15

ACCESSION for

CFSTI WHITE SECTION ☒

DDC BUFF SECTION ☐

UNANNOUNCED ☐

JUSTIFICATION ....

....

BY ....

DISTRIBUTION/AVAILABILITY CODES

DIST.	AVAIL. and/or SPECIAL
1	

INFORMATION SYSTEM DESIGN IN LARGE SCALE LOGISTIC SYSTEMS

E. P. Durbin*
The RAND Corporation
Santa Monica, California

I. INTRODUCTION AND SUMMARY

Operations researchers and systems analysts have become increasingly concerned with information system design. Operations research (O.R.) is careful analysis applied to problems of decisionmakers -- preferably using mathematical models, and has traditionally been concerned with physical production, distribution, and stockage. Operations researchers use techniques such as simulation, Delphi methods, and Operational gaming, and generally aim at finding strategies -- decisions that can be made as functions of variables existing at the time the decision is to be made. O.R., Computer Sciences, and Information Sciences are sometimes confused. O.R. people use computers. Computer Systems designers use O.R., and Information Systems designers use both O.R. and computer designers. The O.R. approach has not been noticeably successful in improving Information Systems Design. This situation is more general in that systems analysis has not been noticeably successful in affecting complex social and environmental problems of resource allocation.

In this paper I will develop the following points:

1. Various factors cause transition to new information systems.
2. Traditional systems analysis recommends top-down design -- "goals to objectives to decision variables to policies to information system specifications."
3. Organizations typically use "bottom up" or "inside out" design.
4. Typically this occurs because of institutional incentives and also because of the complexity of modern systems.

---

*Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of The RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The RAND Corporation as a courtesy to members of its staff. This Paper was prepared for presentation to the NATO Conference, 19-22 May, 1970, Luxembourg.

5. The outcome is typically degraded performance at best.
6. To cope with this reality we must consider incremental or phased development that tries to preserve system design options at each step.

II. DEFINITION OF TERMS

Information Systems

In a complex organization, an information system performs the same function as the nervous system in the human body. This paper is concerned with information systems used by managers and planners in very large organizations. Such systems may be as simple as item stock level reports in a chain of warehouses or as complex as systems that come into play when an expensive spare part is required by an out-of-commission aircraft at a remote airfield. Stockage, airlift, and procurement information as well as repair computations may be required to determine the point of origin for resupply of the required part. A typical logistics information system consists of several complexes of computers tied together with owned or leased communication facilities. Logistics managers may interact with the information system in making daily decisions, and may enter their decisions back into the information system.

Information system concepts develop only slowly. Most attention has been directed at transition between second and third generation computer equipment. Second generation systems are characterized by IBM 7090- or 7094-type equipment, tape units, sequential batch processing, and only one user at a time on the CPU. Third generation systems are characterized by IBM 360/65 series computer, direct access memory, terminal options, multi-programming, and multi-user time sharing.

Transition between computer generations involves shifting transactions from one system to another, perhaps only through software or hardware, but perhaps also by making policy changes, and perhaps changing from batch to on-line processing.

Several factors may lead to initiation of system change. Workload increases. Each new set of transactions creates more work and leads to time delays in processing. Facilities wear out and require increased maintenance. Fashions in computing change and increased operating flexibility is desired. Speed of hardware (not software) improves and arguments concerning dollar costs of each computing operation are difficult to ignore.

III. TOP-DOWN DESIGN

Top-down design is policy oriented and proceeds from the ideas of constrained optimization. It asks:

What are goals of the organization?

What are its operating policies and its policy options?

What are the information requirements of policies and the interaction between policy returns and system costs?

Top-down design attempts to look at the overall organization, its policies, and their interactions.

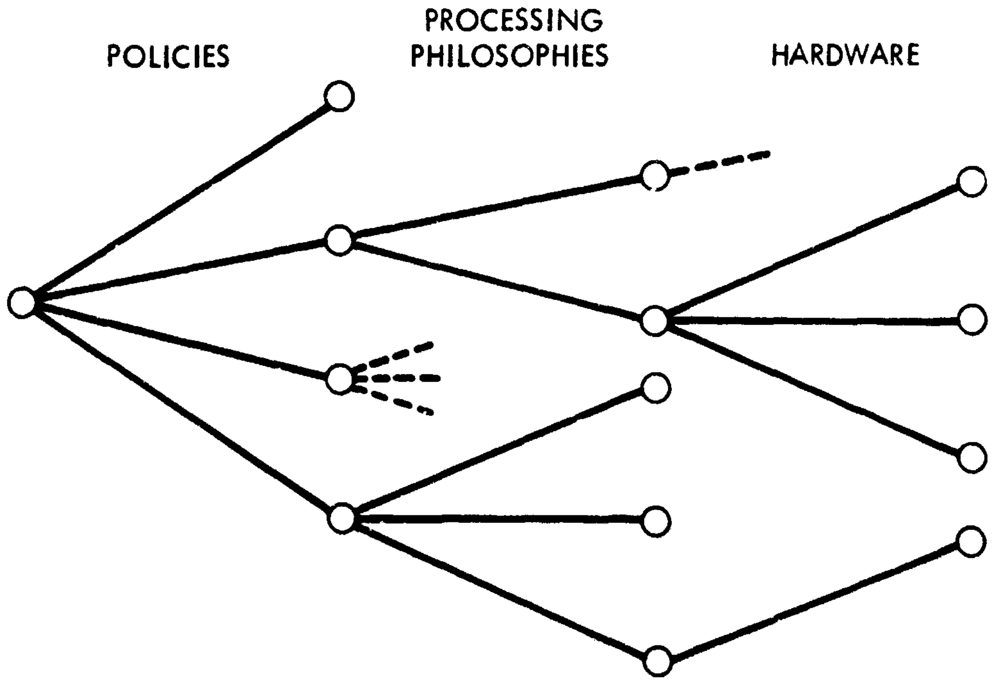
Changes in decision and operating procedures appear to be the source of the major dollar and effectiveness gains in organizations. New technology may be required to implement desired decision and operating procedures, and introduction of new technology may be an essential step. However, resources available for system development are generally limited, often severely. When an initial decision is to take a very large step in introducing new technology, policy improvement will inevitably suffer. The problems associated with simply making new technology run absorb most of the staff. Top-down planning stresses policy and upgrades technology only as necessary. Once a policy base exists, the full range of new technology can be introduced. Processing requirements generate costs. The comparison of policy benefits with processing costs dictates the choice of both policy and information processing schemes. Processing parameters and available technology then lead to hardware selection.

Top-down design typically relies on analytic decision procedures. Such planning emphasizes decision procedures to avoid trouble rather than ad hoc procedures to get out of trouble. Analytic models, simulation, and cost-effectiveness analyses are used to evaluate the worth of policy improvements.

Expending resources on modeling and experimentation requires a tradeoff between time and uncertainty. The more effort put into experimentation and analysis, the greater the reduction in uncertainty about the performance of the ultimate system and reduction in the consequent risk that it will be inadequate. The less effort put into experimentation and analysis, the faster a system gets designed and implemented; but with more attendant uncertainty and risk about ultimate performance. Obviously, when time is available, simulation can be of great benefit.

Laboratory Problem II was conducted in the Logistics Systems Laboratory of The RAND Corporation in 1958-1959, before the ICBM force was fully operational. This laboratory environment allowed the information system to be exposed clearly. Design from the beginning was aimed at filling managerial as well as "housekeeping" needs, and the data base and processing scheme turned out to be flexible in meeting a wide variety of information demands. In the early stages of this simulation the role of the information system appeared to be simply that of serving as the necessary "nerves" of the simulation, under the assumption that the people manning key functions would be able to perform their tasks with reasonable efficiency and dispatch. Not too surprisingly, this skeletal information system showed early signs of providing some useful, and as it turned out, essential feedback of operating results to management. The potential contribution of the information to overall management control of the simulated organization seemed then to justify greater attention to the technology of information generation and processing. LP-II showed systems should be given an integrated design that will be able to adapt to changing demands over time. If the initial definition of data to be captured is satisfactorily broad, meeting new demands can be simply a matter of rearranging or reorganizing data already captured. Not all the operating stresses on an organization can be predicted in advance and decisionmaking and information requirements become clearer with developing experience. Where full-scale operating experience is not available, simulation is a useful tool for exploring the operational, decisionmaking, and information requirements of future organizations, and for developing integrated approaches to their management.

Top-down design can be likened to a complex decision tree with successive branches in Policy, Processing Philosophies, Hardware Configuration.



Policies might include decision procedures for Stock Management, Distribution, Industrial Repair Scheduling, Procurement Policy, or EOQ Purchases. Processing Philosophy must consider the degree of man-machine interaction, the mix of batch or on-line processing types of data base, file management systems, degree of system autonomy or manual override. Hardware considerations must include the tremendous range of equipment available.

Very little is known about where to draw the line -- to stop experimenting and analyzing and start implementing. Some analysis and experimentation is necessary, but we can only base our opinions on where to stop on subjective estimates of the utility of additional research.

The typically recommended general approach has been to:

(1) Develop formal simulation models of the entire system under consideration to use in evaluating alternative policies;

(2) Develop detailed simulation or analytical models of the proposed software (data management and multiprogramming systems for example) and experiment with these. For example to see how the systems behave against different input distributions and to study the trade-offs between data redundancy and information retrieval times.

(3) Develop both gross and detailed simulation models for use in cost/effectiveness and decision-rule studies. These models are elaborations of the system models first developed. They incorporate more of the functional details developed during the system requirements determination and system integration phases and the computer processing details developed during the software requirements development phase. Thus, they are able to attach costs to specific procedures and processing methods and evaluate the benefits achieved through their use.

(4) Survey similar industrial and military systems and collect statistics on software performance. Determine what overhead factors are being incurred and how existing multiprogramming monitors work.

(5) Finally select software structure.

Top-down design is thus characterized by a strong degree of sequential decisionmaking based on improved information.

IV. BOTTOM-UP (INSIDE OUT) DESIGN

The opposite of the top-down approach is "bottom up" or "inside out" analysis. It starts with arbitrary decisions at some detailed level, or decisions about specific policies of the system. It may have a detailed model of one part of the system, and by a process of addition, builds to an overall system picture. As an example, equipment modernization in corporate data processing has led from 407 punch card equipment to 704 computers to 7090 computers to 360/65 computers with no change in processing rules or frequency of interaction with other transactions.

Bottom-up design forces low-level decisions in restricted contexts. It is characterized by arbitrary selection of hardware, and organization policies are set before any overall planning or cost/effectiveness studies are undertaken. Important policy decisions are made without evaluation of their consequences. Bottom-up decisions generally reflect a desire to utilize new and perhaps attractive computer

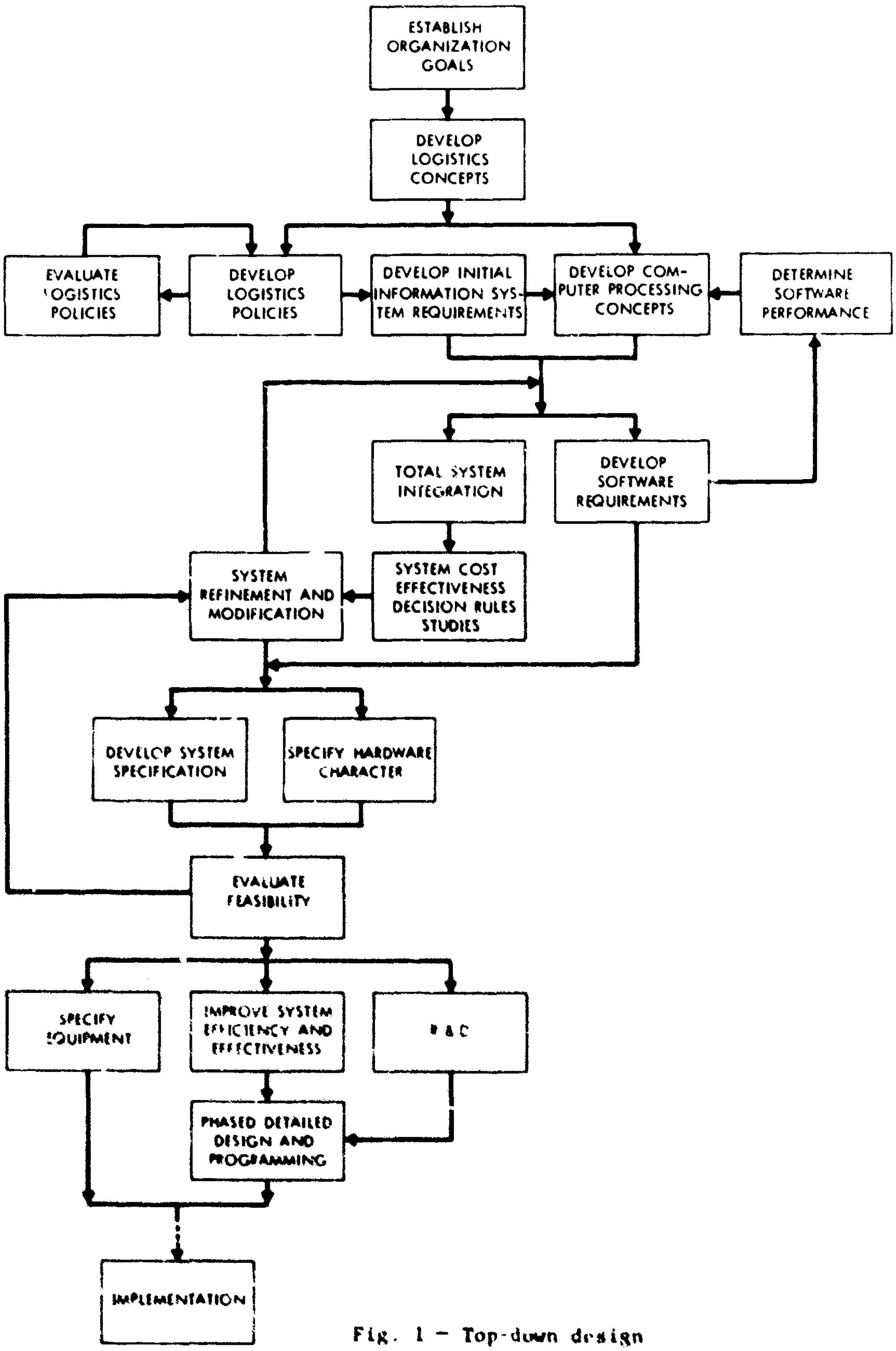


Fig. 1 — Top-down design

hardware rather than a commitment to improvement in the total organization's performance. Thus policy innovations receive only marginal attention, and prime enthusiasm tends to be directed at modernizing the processing equipment. After this initial conceptual set, system design effort must tend toward system operating feasibility rather than system performance or cost.

Bottom-up design is frequently characterized by strong parallel structure or concurrency. There is simultaneous choice of management policy and hardware configuration, and software must then bridge a possibly unbridgeable gap. For example, an Industrial Repair Activity may be provided with consoles or terminals to provide near real time access to data files. But operating decisionmakers may only need to update repair schedules weekly. Real improvements might have come about through provision of additional information to decisionmakers, or provision of some computational or simulation capability on-line.

In summary, design actually observed in reality appears to initially emphasize estimate system hardware requirements. Later emphasis is on modification to achieve feasibility rather than on design exploration and experimentation to improve organizational performance.

V. WHY DOES BOTTOM-UP RATHER THAN TOP-DOWN DESIGN OCCUR?

Bottom-up design is simpler. Top-down design requires determination of organizational and policy goals which are difficult to obtain. It is difficult to model policy effects and interactions. Moreover, bottom-up design rapidly eliminates uncertainty and yields a straightforward implementation plan on paper which is easy to monitor. The arbitrariness is unnoticed until too late.

Management is generally not involved in system design. Technicians are typically in actual charge, and it cannot be assumed that the organization's data processing professionals understand corporate managers' responsibilities. The situation was summarized 100 years ago by a British political commentator who said:

> If left to itself any bureau or department will become technical, self-absorbed, and self-multiplying. It will be likely to overlook the end in the means; it will fail from narrowness of mind; it will be eager in seeming to do; it will be idle in real doing.

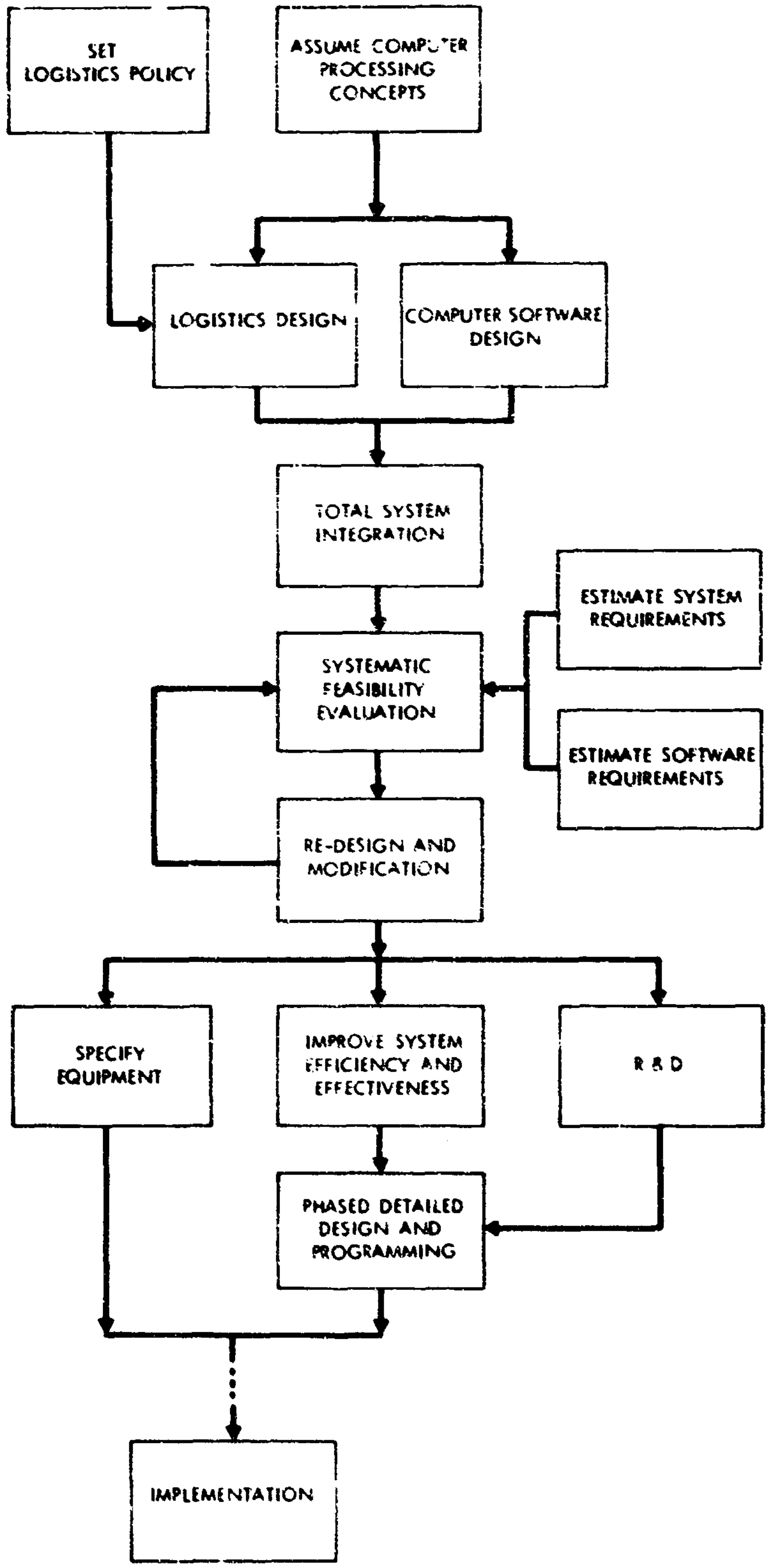


Fig. 2 — Bottom-up design philosophy

This is the case of most in-house data processing departments when it comes to overall information system design. Several organizations or divisions may be involved as users of a new system and it is easier for the design organization to prepare equipment specifications than to assess management needs of diverse organizations.

Procurement mechanisms in large organizations are tedious and system specifications are sometimes desired rapidly. "Buying in" ahead of other capital expenditures in the organization may be necessary. Speed drives the designer to concurrency in policy and hardware selection. This requires development of general purpose software which can be independent of the machines. But such software may itself be difficult to develop or probably contains a small subset of standard languages.

Bottom-up design flourishes because costs and performance evaluation of system specifications is difficult. The performance of a computer is not determined by either the hardware or the software alone. The performance of an installation (hardware, software, and procedures) depends strongly, and sometimes very markedly, on the hardware configuration. Computing standards do not exist for many areas. Metrics have not been identified or established. Costing, especially as it relates to procurement, does not reflect true consumption of resources. Because of all these factors the technical evaluation process is sometimes weak, and lags behind the complexity of systems.

Thus, top-down design is not usually used because it is expensive in dollars, time consuming, and may lead to loss of momentum, and short time schedules for system implementation frequently preclude planning efforts.

VI. DANGER OF BOTTOM-UP DESIGN

Arbitrary schedules, policies, development paths, and system boundaries lead to independent inclusion of different policies at different time points without knowledge of their interactions and implications. This may stunt creativity of designers since there is no overview of flow diagram, and no ability to predict interactions. There is no formal way to introduce new policy, and no valid means of predicting or evaluating policy or system performance.

Rapid pruning of the decision tree in policy and HW configuration occurs, leaving software to fill the gap. This may lead to

Infeasibility: For example, a management evaluation system may provide no data. Or there may be no interfaces between transactions. Or a communication system may break down under heavy volume.

Performance degradation and system rigidity: After implementation an entire information system cannot be altered. Therefore fewer applications may be run, or less frequent updating may be permitted.

Schedule slippages may occur in constructing software or hardware specifications, or in equipment deliveries, or in development of feasible policy applications.

Increased equipment cost may be incurred if extra equipment is required to permit minimal system performance.

In 1958 the U.S. Air Force recognized a need for faster, more responsive information at base level, and began development of centralized data systems for better management of supply, financial services, maintenance, and personnel. There was very little prior study. Designers arbitrarily chose operating goals and available hardware. They were beset by problems which retarded progress and narrowed achievements. The proposal deadlines were very short, and inhibited any substantial innovation. Innovations were further inhibited by the emphasis placed early in the program on showing savings. The project changed from developmental to operational at an early point, thereby diverting resources from development to maintenance. The objectives of the system changed often while computer programming was going on, thereby keeping the project in a continual state of revision and causing schedule slippage.

Thus a vicious circle existed. The lack of study plus the short time schedule led to equipment and policy selection early, necessitating later policy changes which slipped the schedule. In evaluating this effort, the Air Force concluded that existing system experience and knowledge had been inadequate. There were no well defined and effective system development approaches and few adequate techniques for analysis and design. More knowledge was required about the nature of base activities and the hardware requirements for handling them.

Fortunately, the Air Force recognized its needs and took steps to remedy the situation. The early experience pointed up the significant need to initiate data system development projects directed specifically toward improving knowledge. This occurred over 10 years ago. Technology has advanced so rapidly that many of the problems have been carried forward into present day operations. Only continuous effort by the Air Force has kept it from relearning these lessons.

VII. WHAT CAN A CONSULTANT ORGANIZATION CONTRIBUTE IN THIS SITUATION?

It can recognize that, in general, top-down design will not be accomplished unless a relatively long time for development exists, and an excellent consultant group is on hand. In cases where the organization has had large information system experience, the design team will probably use an incremental approach rather than a top-down design approach.

Since the modern information system is complex beyond intuition, the consultant must realize that simple historical examples and homilies will not work. Simple criticism will be ignored. Specific implementable suggestions are required, and specific citations and demonstrations of infeasibility are required.

The consultant must emphasize design for flexibility and change, as well as continuing to advocate modeling and analysis. Our empirical studies and observations of past development projects lead us to believe that a highly phased development strategy is preferred. Rather than allocate available resources across many subsystems focusing resources on one or several critical subsystems has several advantages. First, subsystems are available in the shortest possible time. This strategy permits a staff of modest size and thus a higher quality level can be maintained. Subsystems that appear later in the effort can profit from learning, further policy development, field tests, and simulation exercises. Management and control of phased development are easier since managers do not have to make decisions and follow progress in as many concurrent efforts. Phasing also allows management to recognize that areas differ in terms of (a) payoff, (b) amount of prior work, and (c) ease of development. Phasing does present some difficulties. Some parts of the system must be redesigned and reprogrammed but evidence suggests that the total cost of the phased approach is lower. The real danger of the phased approach is cancellation or loss of momentum prior to completion. Resolution of this problem depends on the organization's procurement policy and on the role taken by top management.

Backup and flexibility must be pressed as key factors. Development is difficult and uncertain. Since management systems always take longer, cost more, and work less well than planned, backup is crucial. Existing systems should be maintained so that they can operate longer if necessary. Buying new equipment that is program compatible with existing equipment provides extensive backup but may be an unavailable or undesirable option for other reasons. Adequate backup gives the development manager important flexibility. If he encounters a need for modification or additional testing that will delay his program, he can make his decision on the costs and gains involved rather than being forced to meet the schedule. Modularity in design can be achieved by selecting equipment to allow rental or purchase add-ons, to change terminal equipment, and number of peripherals, and to change data transmission volumes. Rental flexibility is especially important since they permit return of parts of the system on short notice.

Prototyping portions of the installation should be encouraged. "System" utilization is a misleading phrase if it is not known which part is critical -- memory, communications, or the CPU. One can install, and measure the utilization rate, with actual loads and then add equipment where necessary. Moreover "system" performance in the abstract generally ignores software and staff skills which are observable in the installation.

Management planning is required to produce the system plan and to build the system. Organization is not a final answer to any problem, but it is important that (1) a strong management role be present throughout development to maintain the policymaking or management function, (2) the project be reviewed at top management level to achieve a cross-function view, and (3) the project group include both functional and computer personnel to allow the close interaction needed in modern systems.

SUMMARY

Modern systems analysis is an effort to apply structured rationality to problems of choice. To be of use in Information System design in large organizations the analyst must be aware that techniques of analysis require time and data. Neither may be available. New techniques are required which allow rapid modeling of information systems. We at RAND are developing these. In addition the analyst must understand that institutional factors cause real design to proceed from simultaneous policy and hardware selection through software to the final system. The analyst must supply advice on policy phasing, equipment phasing, flexibility, and backup. This paper has described a situation in which a design process goes backwards from what we suggest. The implications for new analysis techniques may not be so much computational as educational. We begin to view "institutional change" as the main mode of policy change and we must accept risk-avoidance as the primary utility measure of the decisionmaker. As analysts, our emphasis must be on innovation -- attention to creativity-preserving options, and we must pay attention to system design anew rather than to system redesign.